MicroPatent® PatSearch Fulltext: Record 1 of 1

Search scope: JP (bibliographic data only)
Years: 1836-2006
Patent/Publication No.: ((JP09146842))

JP09146842 A
STORAGE SUBSYSTEM
HITACHI LTD

[no drawing]

Abstract:
PROBLEM TO BE SOLVED: To provide the storage subsystem with a controller which eliminates exclusive control over a cache between plural controllers sharing the cache. SOLUTION: The cache areas of caches 33 and 43 in which data are written mutually in multiple are divided by processors and the controllers 30 and 40 access only their controller control areas. The cache areas that the controllers use are fixed to eliminate the need for exclusive control between the processors and prevent deterioration in performance due to multiprocessor constitution.

Inventor(s):
KOBAYASHI RIE
MATSUMOTO YOSHIKO
MURAOKA KENJI

Application No. 07300967 JP07300967 JP, **Filed** 19951120, **A1 Published** 19970606

Original IPC(1-7): G06F01208
G06F01208 G06F00306 G06F00306

Current IPC-R	invention	version	additional	version
Advanced	***G06F00306***	*20060101*		
	G06F01208	*20060101*		
Core	**G06F00306**	20060101		
	G06F01208	20060101		

Patents Citing This One (1):

➡ US7032068 B2 20060418 NEC Corporation
Disk cache management method of disk array device

Home Search List First Prev Go to Next Last

For further information, please contact:
Technical Support | Billing | Sales | General Information

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平9－146842

(43)公開日　平成９年(1997)６月６日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　12/08	３２０	7623－5B	Ｇ０６Ｆ　12/08	３２０	
		7623－5B		Ｊ	
		7623－5B		Ｈ	
3/06	３０２		3/06	３０２Ａ	
	３０４			３０４Ｃ	

審査請求　未請求　請求項の数19　ＯＬ　（全 14 頁）

(21)出願番号　特願平7－300967

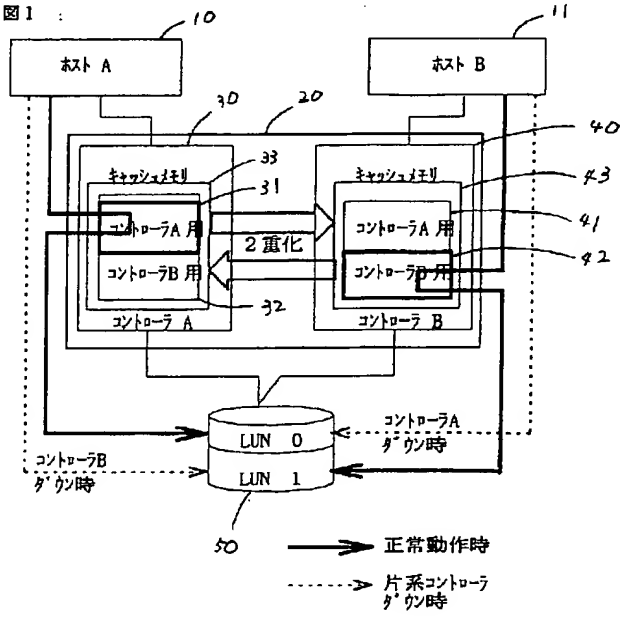
(22)出願日　平成７年(1995)11月20日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　小林　利恵
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内
(72)発明者　松本　佳子
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内
(72)発明者　村岡　健司
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内
(74)代理人　弁理士　小川　勝男

(54)【発明の名称】　記憶サブシステム

(57)【要約】
【課題】キャッシュを共有する複数のコントローラ間のキャシュの排他制御をなくした制御装置を有する記憶サブシステムを提供する。。
【解決手段】互いに多重書きされているキャッシュ３３、４３において、キャッシュ領域を各プロセッサ毎に分割し、各々のコントローラ３０、４０は、自コントローラ制御エリアのみにアクセスする。各コントローラが使用するキャッシュ領域を固定化することにより、プロセッサ間の排他制御を不要とし、複数プロセッサ化に伴う性能劣化を防止することが可能となる。

図１



【特許請求の範囲】
【請求項１】ホストコンピュータのデータを格納し、複数の記憶領域を有する記憶装置と、
該ホストコンピュータの指示に基づいて該記憶装置の制御を行い、該ホストコンピュータと該ディスク装置との間のデータ転送を制御し、該ホストコンピュータと該記憶装置との間を転送されるデータを一時的に保持する複数の領域を有するキャッシュメモリを有する複数のコントローラと前記複数のコントローラ間を接続するパスとを具備する制御装置とを有する記憶サブシステムであって、
前記コントローラには、前記記憶装置の複数の記憶領域のうち少なくとも１つと該コントローラのキャッシュメモリの複数の領域のうち少なくとも１つと前記パスにより接続される他のコントローラのキャッシュメモリの複数の領域のうち少なくとも一つが割当てられることを特徴とする記憶サブシステム。
【請求項２】請求項１記載の記憶サブシステムにおいて、前記コントローラは、前記ホストコンピュータから転送されるデータを該コントローラに割当てられている複数の前記キャッシュメモリに書込むことを特徴とする記憶サブシステム。
【請求項３】請求項２記載の記憶サブシステムにおいて、前記コントローラに障害が発生したときは、前記他のコントローラは該障害コントローラが担当していた前記記憶装置の記憶領域の処理を行うことを特徴とする記憶サブシステム。
【請求項４】請求項３記載の記憶サブシステムにおいて、該他のコントローラはホットスタンバイしているコントローラであって、ホットスタンバイしているコントローラには、キャッシュメモリの記憶領域を割り当てないことを特徴とする記憶サブシステム。
【請求項５】請求項１記載の記憶サブシステムにおいて、前記制御装置は複数の前記コントローラ間を接続するパスを有し、前記コントローラが他のコントローラに割当てられた前記記憶装置の記憶領域に対する処理要求をホストコンピュータから受取ったときは、前記コントローラは、前記他のコントローラに該処理要求を通信することを特徴とする記憶サブシステム。
【請求項６】請求項１記載の記憶サブシステムにおいて、前記キャッシュ領域の分割は、コントローラの負荷に応じて変更することを特徴とする記憶サブシステム。
【請求項７】ホストコンピュータのデータを格納する複数の論理ボリュームを有する磁気ディスクと、
該ホストコンピュータと該ディスク装置との間を転送されるデータを一時的に保持する複数の領域を有するキャッシュメモリと、前記キャッシュメモリとが接続され、該データのデータ転送を制御するデータ転送制御部とを有する複数のコントローラと、複数のコントローラ間を接続するパスとを有し、該ホストコンピュータの指示に基づいて該磁気ディスク装置の制御を行う制御装置とを有する記憶サブシステムであって、
前記コントローラには、前記磁気ディスク装置の複数の論理ボリュームのうち少なくとも１つと該コントローラのキャッシュメモリの複数の領域のうち少なくとも１つと、他のコントローラのキャッシュメモリの複数の領域のうち少なくとも１つとが割当てられることを特徴とする記憶サブシステム。
【請求項８】請求項７記載の記憶サブシステムにおいて、前記コントローラは、前記ホストコンピュータから転送されるデータを該コントローラに割当てられている該コントローラのキャッシュメモリの領域と、該コントローラに割当てられている他のコントローラのキャッシュメモリの領域とに書込むことを特徴とする記憶サブシステム。
【請求項９】請求項８記載の記憶サブシステムにおいて、前記コントローラに障害が発生したときは、前記他のコントローラは該障害コントローラが担当していた前記論理ボリュームの処理を行うことを特徴とする記憶サブシステム。
【請求項１０】請求項９記載の記憶サブシステムにおいて、前記他のコントローラはホットスタンバイしているコントローラであって、、ホットスタンバイしているコントローラには、キャッシュメモリの記憶領域を割り当てないことを特徴とする記憶サブシステム。
【請求項１１】請求項７記載の記憶サブシステムにおいて、前記コントローラが他のコントローラに割当てられた論理ボリュームに対する処理要求をホストコンピュータから受けとったときは、前記コントローラのデータ転送制御部は、該他のコントローラに前記第一のパスを介して処理要求を転送し、該処理要求を受領した該他のコントローラが該論理ボリュームに対する処理を行い、処理結果を、前記コントローラに転送することを特徴する記憶サブシステム。
【請求項１２】請求項７記載の記憶サブシステムにおいて、前記キャッシュ領域の分割は、コントローラの負荷に応じて変更することを特徴とする記憶サブシステム。
【請求項１３】請求項７記載の記憶サブシステムにおいて、前記コントローラ間のパスは、２つの前記コントローラを接続する第一のパスと、、該２つのコントローラの組を接続する第二のパスを含むことを特徴とする記憶サブシステム。
【請求項１４】請求項１３記載の記憶サブシステムにおいて、前記制御装置にコントローラを増設するときは、前記コントローラの２台単位に増設することを特徴とする記憶サブシステム。
【請求項１５】ホストコンピュータのデータを格納し、複数の記憶領域を有する記憶装置と、
該ホストコンピュータの指示に基づいて該記憶装置の制御を行い、該ホストコンピュータと該記憶装置との間の

データ転送を制御し、該ホストコンピュータと該記憶装置との間を転送されるデータを一時的に保持する複数の領域を有するキャッシュメモリを有する複数のコントローラと前記複数のコントローラ間を接続するパスとを具備する制御装置とを有する記憶サブシステムであって、前記コントローラには、前記記憶装置の複数の記憶領域のうち少なくとも１つと該コントローラのキャッシュメモリの複数の領域のうち少なくとも１つと前記パスにより接続される他のコントローラのキャッシュメモリの複数の領域のうち少なくとも一つが割当てられ、
前記ホストコンピュータから転送されるデータは、該コントローラに割当てられている該コントローラのキャッシュメモリの領域と、該コントローラに割当てられている他のコントローラのキャッシュメモリの領域に書込まれることを特徴とする記憶サブシステム。
【請求項１６】請求項１５記載の記憶サブシステムにおいて、前記コントローラに障害が発生したときは、前記他のコントローラは該障害コントローラが担当していた前記記憶装置の記憶領域の処理を行うことを特徴とする記憶サブシステム。
【請求項１７】請求項１５記載の記憶サブシステムにおいて、該他のコントローラはホットスタンバイしているコントローラであって、ホットスタンバイしているコントローラには、キャッシュメモリの記憶領域を割り当てないことを特徴とする記憶サブシステム。
【請求項１８】請求項１５記載の記憶サブシステムにおいて、前記制御装置は複数の前記コントローラ間を接続するパスを有し、前記コントローラが他のコントローラに割当てられた前記記憶装置の記憶領域に対する処理要求をホストコンピュータから受取ったときは、前記コントローラは、前記他のコントローラに該処理要求を通信することを特徴とする記憶サブシステム。
【請求項１９】請求項１５記載の記憶サブシステムにおいて、前記キャッシュ領域の分割は、コントローラの負荷に応じて変更することを特徴とする記憶サブシステム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、上位装置からの情報の入出力要求を制御する制御装置を有する記憶サブシステムに関し、特に、制御装置内のコントローラ及びキャッシュメモリを冗長構成とする記憶サブシステムに関する。
【０００２】
【従来の技術】コントローラ及びディスク等の記憶装置に冗長性を持たせた記憶サブシステムとして、一方の系が現用系として、他方の系が予備系として稼働する２重の系で構成される記憶サブシステムがある。
【０００３】特開平４－２１５１４２に記載されている記憶サブシステムは、現用系のディスク装置の記憶情報を両系からアクセス可能な共用ディスク装置を介して予備系のディスク装置に複写すること、あるいは、現用系コントローラ障害時は、予備系のコントローラによって、現用系のディスク装置の記憶情報を抽出可能とすることによって、コントローラ及びディスク装置障害時のデータ保全性の向上を計っている。
【０００４】
【発明が解決しようとする課題】最近の市場動向として、記憶装置の高性能化、大容量化、低価格化の要求が高まっており、ＲＡＩＤの技術が重視されている。ＲＡＩＤの技術を適用した記憶サブシステムにおいては、複数のディスク装置をアレイ状に構成する。そして、データ書き込み時には、書き込みデータに加えて冗長データを書き込みデータを格納したディスク装置とは異なるディスク装置へ書き込む。アレイ構成内の任意のディスク装置の故障に対しは、他のディスク装置のデータと前記冗長データから障害ディスク装置上のデータを修復可能とすることによって、ディスク装置のデータの保全性の向上を計っている。
【０００５】しかし、ＲＡＩＤの技術を適用した記憶サブシステムは、データの保全性が向上する反面、上述した冗長データ生成／書き込みのために処理時間が増大するため、ホストからのＩ／Ｏ処理と同期して冗長データの生成／書き込みまでを行うと、ホストからのライト性能が大幅に劣化する。従って、ＲＡＩＤの技術を適用した記憶サブシステムのコントローラには、ライトキャッシュが不可欠となっている。
【０００６】ライトキャッシュとは、コントローラ内に搭載された、データを一時的に書き込むキャッシュであり、ホストからのライト要求では、このキャッシュに書き込みを行った時点で、ホストに終了報告を返す。そして、ホストのＩ／Ｏ処理とは非同期に冗長データの生成、ライトデータ及び冗長データのディスク装置への格納を行うことにより書き込み処理時の性能低下を防ぐ。しかし、ライトキャッシュを用いると、データをキャッシュ上に書き込んだ時点でホストに終了報告をするため、キャッシュ上にディスク装置未反映のホストデータが存在する。従って、キャッシュに冗長性がなければ、キャッシュ障害時に、ユーザデータロストとなる。したがって、特にデータの高信頼性が強く求められる記憶サブシステムに用いる制御装置では、従来のコントローラ、記憶装置の冗長構成に加え、一般的にキャッシュにも冗長性を持たせることが行われている。
【０００７】コントローラを多重化した記憶サブシステムにおいて、単にキャッシュを多重化すると、キャッシュ上のデータを複数の制御装置から同時にアクセスすることによるデータ整合性矛盾を防ぐためにキャッシュアクセス時に複数の制御装置からの排他制御が必要になる。そして、コントローラを多重化した記憶サブシステムでは、この排他制御により、シングルコントローラの

記憶サブシステムに比べて性能が低下する。

【０００８】本発明の目的は、コントローラの多重化及びキャッシュの多重化に伴う、プロセッサ間のキャッシュの排他制御を無くし、性能を落とすことなく信頼性を上げることにある。

【０００９】

【課題を解決するための手段】上記の目的を達成するため、本発明による記憶サブシステムは、各プロセッサ毎に処理担当の論理ボリュームを排他的に決める手段と、あるプロセッサが受領したホストコンピュータからの要求が、担当外であった場合は、担当プロセッサに処理要求を通信する手段と、上記通信を受領したプロセッサは、処理結果を要求元プロセッサに通信する手段と、各プロセッサ毎に、ディレクトリ／データセグメント等のキャッシュ構成要素を持つ手段と、上記構成要素の状態をプロセッサの負荷に応じてダイナミックに変更する手段と、ホストコンピュータからのライトデータを複数のコントローラ上のキャッシュへ多重書きする手段と、コントローラ障害時には、障害コントローラ内プロセッサの持つキャッシュ構成要素の制御権を正常系コントローラ内のプロセッサに切り替える手段と、コントローラ復旧時には、該制御権を復旧プロセッサに戻す手段と、ディスク装置への書き込み時にキャッシュメモリ障害が発生した際は、多重書きしている他キャッシュからディスク装置に書き込みを行う手段とを有する。

【００１０】上述した手段により、複数のプロセッサ間で、キャッシュを排他制御することなく複数のコントローラ上のキャッシュへ多重書きを行うことができ、複数プロセッサ化に伴う性能低下の発生を防ぎ、性能を落とすことなく信頼性の向上を計ることができる。

【００１１】また、上述の手段により、キャッシュメモリ障害時には、多重書きしている他キャッシュからディスク装置への書き込みを行い、データロストを防止できる。

【００１２】さらに、上記手段により、コントローラ障害時には、自動的に、正常系に切り替えて処理続行が可能であり、また、コントローラ復旧時には、自動的に、復旧系に処理を戻すことが可能となり、システムの無停止運用を実現できる。

【００１３】

【発明の実施の形態】図１は、本発明の概念図である。

【００１４】図１において、１０、１１はホストコンピュータ、２０はデュアルコントローラ構成をとる制御装置、５０はディスク装置であり、ディスク装置５０は、論理ボリューム０と論理ボリューム１の２つの論理ボリュームに分割されている。

【００１５】ホストＡ１０は、制御装置２０内のコントローラＡ３０を介して、論理ボリューム０の処理を行っており、ホストＢ１１は、制御装置２０内のコントローラＢ４０を介して、論理ボリューム１の処理を行っている。

【００１６】ここで、コントローラＡ３０には論理ボリューム０が、コントローラＢ４０には論理ボリューム１が処理担当論理ボリュームとして割当てられている。又、コントローラ内のキャッシュの領域は、それぞれ、コントローラＡ用キャッシュ３１、４１、コントローラＢ用キャッシュ３２、４２に２分割されている。そして、コントローラＡ用キャッシュ３１と４１の間で２重書きを行い、又、コントローラＢ用キャッシュ３２と４２の間でも２重書きを行う。

【００１７】コントローラA.３０は、通常、コントローラＡ用キャッシュ３１と４１を用いて、Ｉ／Ｏ処理を行い、同様に、コントローラＢ４０は、コントローラＢ用キャッシュ３２と４２を用いて、Ｉ／Ｏ処理を行う。このように、コントローラ毎に使用するキャッシュ領域を個別に割り当てることにより、コントローラ間の排他制御を無くし、コントローラ台数増加に伴う性能劣化を防ぐことができる。

【００１８】また、コントローラＢ４０障害時には、コントローラＢ用キャッシュ３２、４２をコントローラＡ３０が使用することにより、ホストＡ１０からコントローラＡ３０を介して、コントローラＢ４０の処理担当であった論理ボリューム１への処理を続行させることができる。

【００１９】以下、本発明によるマルチコントローラ構成の制御装置の１実施例を図面を用いて説明する。

【００２０】図２は、本発明をマルチコントローラ構成の磁気ディスクアレイサブシステムに適用した場合の構成図である。

【００２１】図２において、１０００、１１００、１２００、１３００はデータ処理を行う中央処理装置であるホストコンピュータ、２０００はマルチコントローラ構成をとりディスク装置の制御を行う制御装置、７０００，７１００はホストコンピュータのデータを格納するディスク装置である。ここで、制御装置２０００は、ホストバスに直結したスロットに差し込みホスト筐体内に組み込む場合もあるし、制御装置として独立した筐体に組み込む場合もあるし、ディスク装置を組み込んだ筐体として実現する場合もある。また、ディスク装置群７０００及び７１００は、データディスクとパリティディスクからなるパリティグループを含んでいる。さらに、ディスク装置群７０００は、論理ボリューム０と論理ボリューム１とに、ディスク装置群７１００は論理ボリューム２と論理ボリューム３とに分割されている。

【００２２】制御装置２０００は、ホストコンピュータ１０００、１１００とディスク装置７０００間のデータ転送を制御するコントローラ３０００、４０００及びホストコンピュータ１２００、１３００とディスク装置７１００間のデータ転送を制御するコントローラ５０００、６０００より構成される。

【００２３】コントローラ３０００は、ホストコンピュータ１０００とのプロトコル制御を行うホストＩ／Ｆ制御部３１００、コントローラ全体を制御するマイクロプロセッサ（以下「プロセッサ」という。）３２００、データの転送を実行するデータ転送制御部３３００、ホストコンピュータ１０００とディスク装置７０００のデータ転送時及びプロセッサ間通信時に用いられるキャッシュ３４００、各ディスク装置７０００とのプロトコル制御を行うＤＲＶＩ／Ｆ制御部３５００より構成される。コントローラ４０００、５０００、６０００はコントローラ３０００と同一の構成である。

【００２４】プロセッサ３２００は、後述の手段により、あらかじめプロセッサ毎に排他的に割り当てた担当論理ボリュームの処理を行う。このプロセッサ毎の担当論理ボリュームの指定は、ホストコンピュータから論理ボリューム毎の担当プロセッサ指定コマンドを受け取ることにより、ダイナミックに設定可能である。このプロセッサと担当論理ボリュームとの対応情報は、後述のキャッシュ上の共通メモリ領域３４１０、４４１０に格納する。

【００２５】データ転送制御部３３００はプロセッサ３２００からの指示により、ホストコンピュータ１０００からのライトデータを指定キャッシュに多重書きする機能を備えている。この実施例の構成では、キャッシュ３４００とキャッシュ４４００の間で２重書きを行い、また、キャッシュ５４００とキャッシュ６４００の間でも２重書きを行う。以下、キャッシュ３４００とキャッシュ４４００の２面に２重書きする方式について説明する。

【００２６】キャッシュ３４００とキャッシュ４４００の内容について図３を用いて説明する。尚キャッシュ３４００とキャッシュ４４００は内部構成が同一であるため、キャッシュ３４００を例に説明する。キャッシュ３４００は、プロセッサ間通信に用いる制御情報を格納している共通メモリ領域３４１０、プロセッサ３２００用領域３４８０、プロセッサ４２００用領域３４９０より構成される。

【００２７】プロセッサ３２００用領域３４８０は、ホストコンピュータとディスク装置間のデータ転送時、データを１次的に格納するデータ格納エリア３４８２、データ格納エリア３４８２を管理するデータ管理情報３４８１より構成され、データ格納エリア３４８２に格納するライトデータと、このライトデータの管理情報は、キャッシュ４４００内のプロセッサ３２００用領域４４８０に２重書きを行う。同様に、プロセッサ４２００用領域３４９０は、プロセッサ４２００により、キャッシュ４４００内のプロセッサ４２００用領域４４９０のライトデータとライトデータの管理情報が２重書きされている。

【００２８】共通メモリ領域３４１０は、論理ボリューム担当プロセッサ情報３４２０、プロセッサ負荷情報３４３０、多重書き情報３４５０、プロセッサ間コミュニケーションメモリ３４６０より構成され、これらの情報は全て、データ転送制御部３３００、４３００によって、キャッシュ３４００と４４００に２重書きされている。

【００２９】図３（ｃ）にプロセッサ間コミュニケーションメモリの構成を示す。プロセッサ間コミュニケーションメモリ３４６０は、プロセッサ３２００、４２００、５２００、６２００毎の書き込み用メモり３４６１、３４６２、３４６３、３４６４より構成される。図３（ｄ）にプロセッサ書き込み用メモリの構成を示す。プロセッサ３２００書き込み用メモリ３４６１は、自プロセッサ以外のプロセッサ４２００、５２００、６２００への要求用エリア３４７１、３４７２、３４７３と自プロセッサ以外のプロセッサ４２００、５２００、６２００からの要求に対する応答用エリア３４７４、３４７５、３４７６より構成される。プロセッサ４２００、５２００、６２００書き込み用メモリ３４６２、３４６３、３４６４の内部構成は、プロセッサ３２００書き込み用メモリ３４６１と同一構成である。

【００３０】キャッシュ５４００とキャッシュ６４００との間も、共通メモリ領域を除いて、キャッシュ３４００とキャッシュ４４００との間と同様に２重化が行われている。共通メモリ領域は、キャッシュ３４００、４４００に２重書きされている情報を制御装置内の全プロセッサで共有するため、キャッシュ５４００、６４００には存在しない。

【００３１】本発明を実施する制御装置では、コントローラの増設はコントローラ２台単位で行い、対になったコントローラのキャッシュ間のみで２重書きを行うとともに、ドライブ側のデータバスについても、それぞれのディスク装置は対になったコントローラにのみ接続することによりハードウェア構成を簡略化し、ドライブ側データバス上の競合を回避することが可能となる。

【００３２】次に本実施例における、磁気ディスクサブシステムでの、ホストコンピュータ１０００からのＩ／Ｏ処理について図４、図５、図６を用いて説明する。まず最初に、プロセッサ３２００担当論理ボリュームへのＩ／Ｏ処理について説明する。

【００３３】図４は、ホストからのＩ／Ｏ処理を示すフローチャートである。ホストコンピュータ１０００からの書き込み要求時、プロセッサ３２００は、まず、共通メモリ領域３４１０内の論理ボリューム担当プロセッサ情報３４２０によって、処理要求論理ボリュームの担当プロセッサ情報を取得し、自処理担当論理ボリューム（ＬＵＮ）への処理かの判定を行い（ステップ９０２）、自プロセッサ処理担当論理ボリュームへの処理であることを認識する。次に、処理種別の判定を行い（ステップ９０３）、書き込み処理であることを認識する。

ホストＩ／Ｆ制御部３１００により、書き込み論理データを受領し、データ転送制御部３３００によってキャッシュ３４００のコントローラ３０００用領域３４８０とキャッシュ４４００のコントローラ３０００用領域４４８０とにその管理情報とともに２重に格納する（ステップ９０４）。そして、この時点でホストコンピュータ１０００に終了を報告する（ステップ９０５）。

【００３４】図５は、キャッシュ内のデータをディスク装置に格納する処理を示すフローチャートである。プロセッサ３２００は、ホストコンピュータ１０００からのＩ／Ｏ処理とは非同期にプロセッサ３２００用領域３４８０上のライトデータをデータ転送制御部３３００とＤＲＶ　Ｉ／Ｆ制御部３５００によりディスク装置群７０００に格納する（ステップ９２２）。この際、キャッシュのメモリ障害により読み込みエラーが発生した場合（ステップ９２３）は、２重化しているプロセッサ３２００用領域４４８０からディスク装置７０００へ格納する（ステップ９２４）ことによりデータ損失を防止することができる。

【００３５】ホストコンピュータ１０００からの読み込み要求時は、プロセッサ３２００は、上記書き込み処理同様、自プロセッサ処理担当論理ボリューム（ＬＵＮ）への処理であることを認識（ステップ９０２）した後、処理種別の判定を行う（ステップ９０３）。Ｉ／Ｏ処理が読み込み処理であることを認識すると、データ転送制御部３３００とＤＲＶ　Ｉ／Ｆ制御部３５００によりデータをディスク装置群７０００からキャッシュ３４００のコントローラ３０００用領域３４８０に格納し（ステップ９０６）、ホストコンピュータに転送する（ステップ９０７）。

【００３６】次にホストコンピュータ１０００からコントローラ４０００担当論理ボリュームへのＩ／Ｏ処理について説明する。

【００３７】ホストコンピュータ１０００からの書き込み要求時、プロセッサ３２００は、まず、共通メモリ領域３４１０内の論理ボリューム担当プロセッサ情報３４２０によって、処理要求論理ボリュームの担当プロセッサ情報を取得し、自処理担当論理ボリュームへの処理かの判定を行い（ステップ９０２）、処理担当外論理ボリュームへの処理であることを認識する。次に、処理種別の判定を行い（ステップ９０８）、書き込み処理であることを認識する。そして、ホストコンピュータ１０００からの書き込み論理データをキャッシュメモリのコントローラ３０００用領域３４８０に格納し、書き込み処理をこの論理ボリュームの担当であるコントローラ４０００へ要求する（ステップ９０９）。

【００３８】プロセッサ３２００は、プロセッサ４２００に書き込み処理を要求するために、、書き込みデータ論理アドレス、書き込みデータのキャッシュ上の格納アドレス、データ長及び処理種別情報をデータ転送制御部３３００により共通メモリ領域３４１０、４４１０内のプロセッサ３２００書き込み用メモリ内のプロセッサ４２００への要求用エリアに２重に格納する。ここで、処理種別情報とは、書き込み処理か読み込み処理かを判断する情報である。プロセッサ４２００は、例えば１０ｍｓといった一定時間で、共通メモリ領域３４１０、４４１０の自プロセッサへの要求用エリアを参照にいき、他プロセッサからの要求を認識する。

【００３９】図６は、プロセッサ３２００からの処理要求を受信したときのプロセッサ４２００の処理を示すフローチャートである。前述の方法により、プロセッサ３２００からの要求を認識（ステップ９３１）したプロセッサ４２００は、プロセッサ３２００書き込み用メモリ内のプロセッサ４２００への要求用エリア内の処理種別を参照し、書き込み処理要求であることを認識する（ステップ９３２）。そして、プロセッサ４２００は、プロセッサ３２００書き込み用メモリ内のプロセッサ４２００への要求用エリア内の書き込み論理アドレス、書き込みデータのキャッシュ上の格納アドレス、データ長を取得し（ステップ９３３）、キャッシュ３４００内の該格納アドレスからデータ長分の書き込みデータをプロセッサ４２００用領域３４９０と４４９０に、その管理情報である書き込み論理アドレスとデータ長と共に、２重に格納する（ステップ９３４）。そして、終了情報を共通メモリ領域３４１０、４４１０内のプロセッサ４２００書き込み用メモリ内のプロセッサ３２００からの要求に対する応答用エリアに設定することにより、プロセッサ３２００に処理終了を通信する（ステップ９３５）。

【００４０】プロセッサ３２００は、プロセッサ４２００に対する処理要求後は、プロセッサ４２００書き込み用メモリ内のプロセッサ３２００からの要求に対する応答用エリアを参照することにより、プロセッサ４２００の処理の終了を監視（ステップ９１０）しており（図４参照）、この処理終了の通信を受けて、ホストコンピュータ１０００に終了を報告する（ステップ９０５）。プロセッサ４２００は、この後、図５に従ってホストＩ／Ｏ処理とは非同期に、この書き込みデータのディスク装置７０００への書き込み処理を行う。

【００４１】図４において、ホストコンピュータ１０００から読み込み要求があったときは、プロセッサ３２００は、上記書き込み要求受領時同様、処理担当外論理ボリューム（ＬＵＮ）への処理であることを認識した（ステップ９０２）後、処理種別の判定を行う（ステップ９０８）。読み込み処理であることを認識すると、プロセッサ３２００は読み込み要求論理アドレス、読み込みデータのキャッシュ上の格納許可アドレス、データ長、処理種別情報を共通メモリ領域３４１０、４４１０内のプロセッサ３２００書き込み用メモリ内のプロセッサ４２００への要求用エリアに格納することにより、該ＬＵＮ処理担当であるプロセッサ４２００に読み込み要求を通

信する（ステップ９１１）。

【００４２】図６において、プロセッサ３２００からの要求を認識（ステップ９３１）したプロセッサ４２００は、共通メモリ領域内の情報により、読み込み処理であることを認識する（ステップ９３２）。そして、共通メモリ領域から読み込み要求論理アドレス、読み込みデータのキャッシュ上の格納許可アドレス、データ長を取得する（ステップ９３６）。次に、データをディスク装置７０００からプロセッサ４２００用領域４４９０に格納し、このデータをキャッシュ３４００上の格納許可アドレスに格納する（ステップ９３７）。さらに、共通メモリ領域３４１０、４４１０内のプロセッサ４２００書き込み用メモリ内のプロセッサ３２００からの要求に対する応答用エリアに終了情報を設定することにより、プロセッサ３２００に読み込み終了を通信する（ステップ９３５）。

【００４３】図４において、プロセッサ４２００の処理終了を監視（ステップ９１２）していたプロセッサ３２００は、この読み込み終了報告を受けて、データをホストコンピュータに転送する（ステップ９１３）。

【００４４】このように、プロセッサ３２００は、通常、プロセッサ３２００用領域３４８０と４４８０を用いて、Ｉ／Ｏ処理を行う。同様に、プロセッサ４２００は、通常、プロセッサ４２００用領域３４９０と４４９０を用いて、Ｉ／Ｏ処理を行う。

【００４５】このように、プロセッサ毎に使用するキャッシュ領域を固定化することにより、プロセッサ間の排他制御を無くし、プロセッサ台数増加に伴う性能劣化を防ぐことができる。特にホストコンピュータ間でファイル（論理ボリューム）をシェアしないシステムにおいては、接続しているコントローラ内のプロセッサにこの論理ボリュームを割り当てておくことにより、Ｉ／Ｏ処理のときのプロセッサ間の通信制御を不要とし、さらなる性能向上を可能とする。

【００４６】次にコントローラ４０００の障害時の自動切り替え／復旧方式について図７、図８を用いて説明する。Ｉ／Ｏ処理実行中、コントローラ４０００の障害を検知したプロセッサは、共通メモリ領域３４１０を用いて、残りの全プロセッサにコントローラ４０００の障害を通信する。この際、コントローラ４０００とキャッシュを２重書きしているコントローラ３０００内のプロセッサ３２００には、処理の引継要求も通信する。本実施例では、プロセッサ３２００が障害を検知した場合について説明する。

【００４７】図７は、プロセッサ３２００がコントローラ４０００の障害を検知した場合のプロセッサ３２００の処理を示すフローチャートである。プロセッサ３２００はＩ／Ｏ処理実行中（ステップ９５０）、コントローラ４０００の障害を検知（ステップ９５１）すると、前述の方法により、プロセッサ５２００、６２００にコントローラ４０００の障害を通信する。そして、障害コントローラをシステムから切り放すため、キャッシュ３４００と４４００へ２重書きされているホストコンピュータからの書き込みデータ及び共通メモリ領域のデータを、キャッシュ３４００への１重書きに変更することをデータ転送制御部３３００に指示する（ステップ９５２）。また、プロセッサ３２００からの要求を認識したプロセッサ５２００、６２００は、共通メモリ領域をキャッシュ３４００への１重書きに変更する。次に、プロセッサ３２００は、プロセッサ４２００の処理を引き継ぐ為に、プロセッサ４２００用領域の制御権をプロセッサ３２００に切り替える（ステップ９５３）。これらの処理により、制御権の切り替えが完了し、プロセッサ３２００は通常のＩ／Ｏ処理を再開する（ステップ９５４）。

【００４８】図８は、障害が発生したコントローラ４０００の復旧処理を示すフローチャートである。コントローラ４０００の障害部位が交換（ステップ９７１）されると、プロセッサ４２００は、共通メモリ領域３４１０を用いて全プロセッサに復旧開始を伝達する（ステップ９７２）。プロセッサ３２００、５２００、６２００は、この復旧開始の伝達を受けて（ステップ９５５）、それぞれのコントローラのデータ転送制御部にキャッシュ３４００と４４００への２重書きを指示すると共に、共通メモリ領域３４１０、４４１０を用いて、処理終了の応答をプロセッサ４２００に通信する（ステップ９５６）。この終了報告を全プロセッサから受領（ステップ９７３）したプロセッサ４２００は、キャッシュ４４００のデータ回復を行う（ステップ９７４）。データ回復が完了すると、共通メモリ領域３４１０、４４１０を用いて、プロセッサ３２００に復旧完了を伝達する（ステップ９７５）。

【００４９】図７において、この完了通知を受けた（ステップ９５８）プロセッサ３２００は、プロセッサ４２００用領域の制御権をプロセッサ４２００に復旧（ステップ９５９）させ、共通メモリ領域を用いて、制御権の復旧をプロセッサ４２００に伝達する（ステップ９６０）。図８において、この伝達を受けた（ステップ９７６）プロセッサ４２００は、Ｉ／Ｏ処理を再開させる（ステップ９７７）。

【００５０】尚、以上の実施例においては、コントローラ毎にプロセッサ、ホストＩ／Ｆ制御部を１つ持った例を示したが、これらの数は任意でも、ホストコンピュータからのコマンドを受け取ったプロセッサが、担当プロセッサに処理要求を伝達することにより、同様に実現できる。

【００５１】また、キャッシュの分割方式は、プロセッサ毎に均等ではなく、ユーザの指定により設定／変更可能である。特に、特定プロセッサをホットスタンバイで動作させる場合には、キャッシュ領域をホットスタンバ

イのプロセッサには割り当てないことにより、キャッシュを有効に利用することができる。又、プロセッサの負荷に応じてダイナミックに変更することも可能である。ユーザの指定により分割を行うか、プロセッサの負荷に応じて変更を行うかの指示は、本実施例では、ホストコマンドにより行うが、パネルといった装置を接続し、そこから入力する形を取っても、むろん良い。

【００５２】つぎに、コントローラのキャッシュの動的割当の実現方式について、以下、説明する。まず、キャッシュの管理方式について、図９を用いて説明する。

【００５３】プロセッサ毎に持つデータ格納エリアは、セグメント９８３と呼ばれる管理単位に分割されている。セグメントは、セグメント毎にセグメント管理ブロック９８１（以下ＳＧＣＢという。）をデータ管理情報内に持ち、セグメントを管理する情報とセグメントアドレスが格納されている。又、これらのＳＧＣＢは、そのセグメントの属性によって、ダーティキュー９８０とクリーンキュー９８２という２つキューに分けられて接続されている。ダーティキュー９８０には、ディスク未反映のライトデータを格納しているセグメントのＳＧＣＢが接続されており、それ以外のＳＧＣＢは、クリーンキュー９８２に接続されている。

【００５４】キャッシュの動的割当を実現するために、プロセッサ毎の負荷情報を共通メモリ領域に持つ。この負荷情報として、例えば、キャッシュ内のクリーンＳＧＣＢ量を用いる。各プロセッサは、ＳＧＣＢのクリーン、ダーティ間のキュー遷移契機に、この情報を更新する。プロセッサは、例えば、１分といった一定周期でこの情報を参照にいき、キャッシュを共有しているプロセッサ内で最も負荷の低いプロセッサのクリーンキューから最も負荷の高いプロセッサのクリーンキューへ、その負荷が同じになるまでＳＧＣＢと管理セグメントを移行させる。この際、使用中のＳＧＣＢは、移行対象外とする。移行の際は、ＳＧＣＢの格納データ情報はクリアする。この移行の間は、プロセッサ通信を用いて、移行を行うプロセッサのＩ／Ｏ処理はとめる。

【００５５】また、以上の実施例においては、２台のコントローラ間でキャッシュを共有し、各々、対コントローラのキャッシュに２重書きする例を示したが、キャッシュ領域がプロセッサ毎に分割されていれば、そのキャッシュの共有化方式、多重書き方式は、任意の方式でも、同様に実現できる。

【００５６】キャッシュ多重書きの例を図１０に示す。（１）は、装置全体でキャッシュを共有しあい、２重書きする方式である。つまり、プロセッサ３２００はキャッシュ３４００、４４００を用いて、プロセッサ４２００はキャッシュ４４００、５４００を用いて、プロセッサ５２００はキャッシュ５４００、６４００を用いて、プロセッサ６２００はキャッシュ６４００、３４００を用いて２重書きを行っている。

【００５７】（２）は、装置全体でキャッシュを共有しあい、全キャッシュに多重書きする方式である。つまり、プロセッサ３２００、４２００、５２００、６２００は、それぞれキャッシュ３４００、４４００、５４００、６４００を用いて、多重書きを行っている。このケースにおいて、コントローラが障害となった場合は、キャッシュを共有しているプロセッサ間でもっとも負荷の低いプロセッサが、障害コントローラ担当論理ボリュームの処理を引き継ぐ。これらのケースにおいては、任意のプロセッサが障害コントローラ担当論理ボリュームの処理を引き継げるように、ディスク側のデータバスを、装置内の全ディスク装置、全コントローラで共通のバスに接続しておく。もちろん、これらの多重書き方式を装置内で混在させることも可能である。これらの多重書き方式の指定は、共通メモリ領域３４１０、４４１０に多重書き情報を持ち、各々のプロセッサ３２００、４２００、５２００、６２００が、この情報を元に、書き込みデータの転送方式をデータ転送制御部３３００、４３００、５３００、６３００に指示することにより実現できる。

【００５８】

【発明の効果】本発明によれば、コントローラ及びキャッシュメモリを２重化した記憶サブシステムにおいて、各コントローラにキャッシュメモリの一部及び論理ボリュームを割り当てることによりキャッシュメモリに対するコントローラ内のプロセッサ間の排他制御が無くなるため、複数プロセッサ化による応答性能劣化を防ぐことができる。

【００５９】また、複数のキャッシュへ多重書きすることにより、キャッシュ障害時には、多重書きしている他キャッシュからディスクに書き込むことができるため、データロストを防ぐことができる。さらに、コントローラ障害時にキャッシュメモリの制御を正常なコントローラに切り替える手段とコントローラ障害から復旧する手段を設けることにより、システムを無停止で運用することができる。

【図面の簡単な説明】

【図１】本発明の概要を表す構成図である。

【図２】本発明の実施例である制御装置の構成図である。

【図３】本発明の実施例であるコントローラのキャッシュの構成を示す図である。

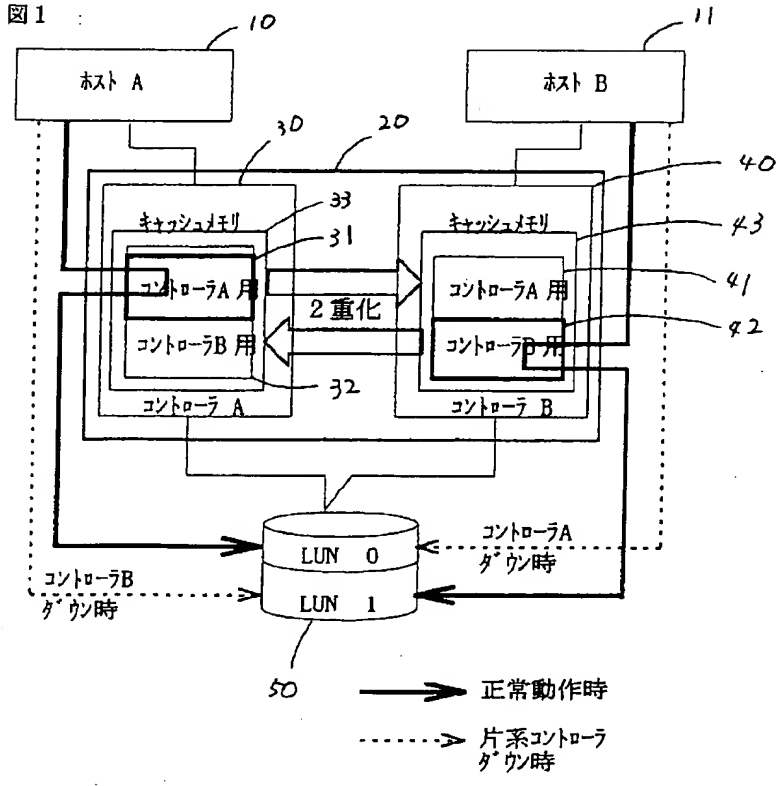
【図４】本発明の実施例によるコントローラのホストからのＩ／Ｏ処理の動作を示すフローチャートである。

【図５】本発明の実施例によるコントローラのキャッシュ内のデータをディスク装置に格納する動作を示すフローチャートである。

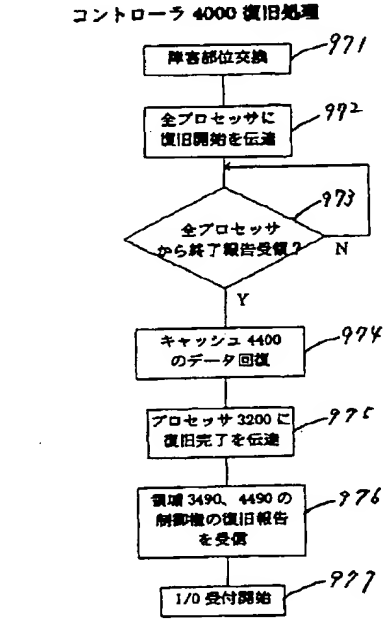
【図６】本発明の実施例による他のコントローラから処理要求を受けとったコントローラの制御装置の動作を示すフローチャートである。

【図７】本発明の実施例による他のコントローラの障害を検出したコントローラの動作を示すフローチャートである。
【図８】本発明の実施例による障害が発生したコントローラの復旧処理の動作を示すフローチャートである。
【図９】本発明の実施例によるコントローラおいて用いられるキャッシュの管理方式を示す図である。
【図１０】本発明の他の実施例二夜コントローラのキャッシュの構成を示す図である。
【符号の説明】
１０／１１：ホストコンピュータ
２０：制御装置
３０／４０：コントローラ
３１／４１：コントローラＡ用キャッシュメモリ
３２／４２：コントローラＢ用キャッシュメモリ
３３／４３：キャッシュメモリ
５０：ディスク装置
１０００／１１００／１２００／１３００：ホストコンピュータ
２０００：制御装置
３０００／４０００／５０００／６０００：コントローラ
３１００／４１００／５１００／６１００：ホストＩ／Ｆ制御部
３２００／４２００／５２００／６２００：マイクロプロセッサ
３３００／４３００／５３００／６３００：データ転送制御部
３４００／４４００／５４００／６４００：キャッシュ
３５００／４５００／５５００／６５００：ＤＲＶＩ／Ｆ制御部
７０００／７１００：ディスク装置群

【図１】

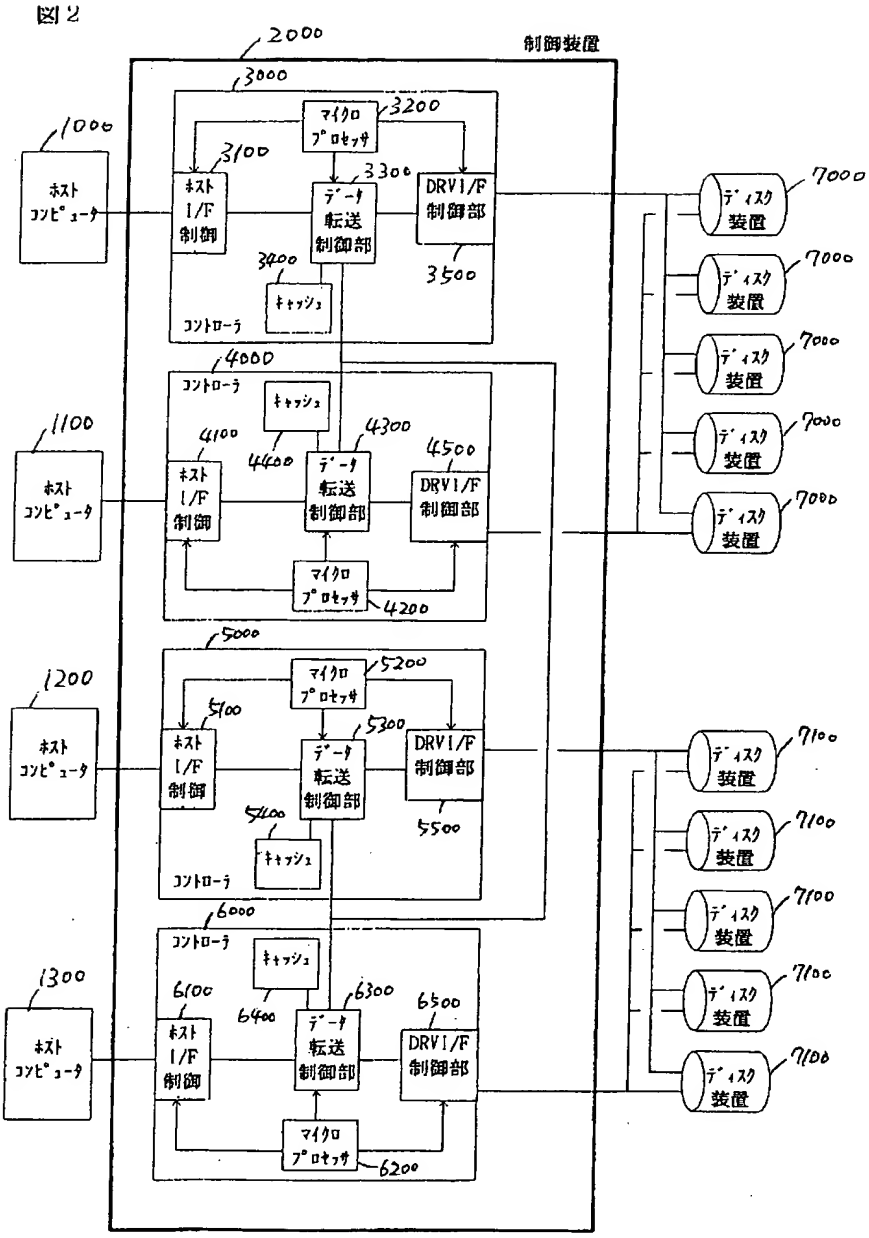


【図８】



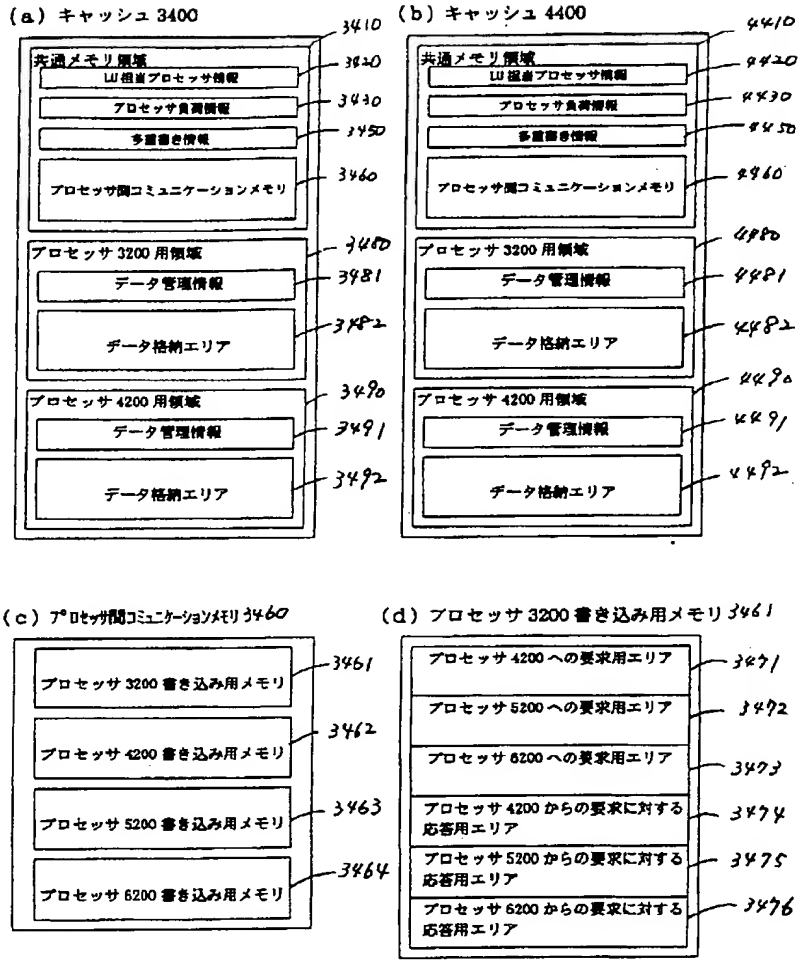
【図２】

図２


2000
制御装置
3000
マイクロプロセッサ
3200
1000
ホストコンピュータ
3100
ホストI/F制御
3300
データ転送制御部
DRVI/F制御部
3500
3400
キャッシュ
コントローラ
7000
ディスク装置
4000
コントローラ
キャッシュ
1100
ホストコンピュータ
4100
4400
4300
データ転送制御部
4500
DRVI/F制御部
マイクロプロセッサ
4200
5000
1200
ホストコンピュータ
マイクロプロセッサ
5200
5100
ホストI/F制御
5300
データ転送制御部
DRVI/F制御部
5500
5400
キャッシュ
コントローラ
7100
6000
コントローラ
キャッシュ
1300
ホストコンピュータ
6100
6400
6300
データ転送制御部
6500
DRVI/F制御部
マイクロプロセッサ
6200

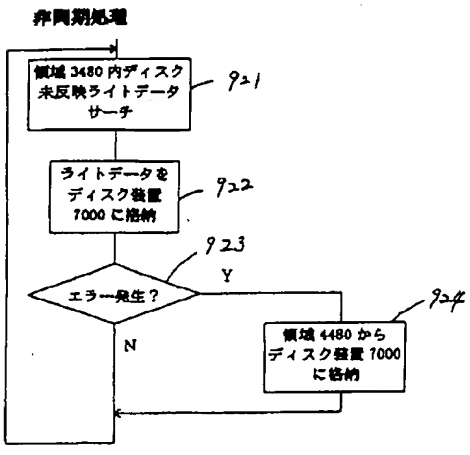
【図３】

図３



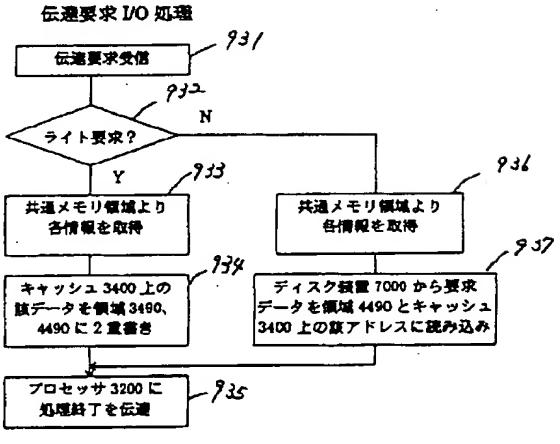
【図５】

図５



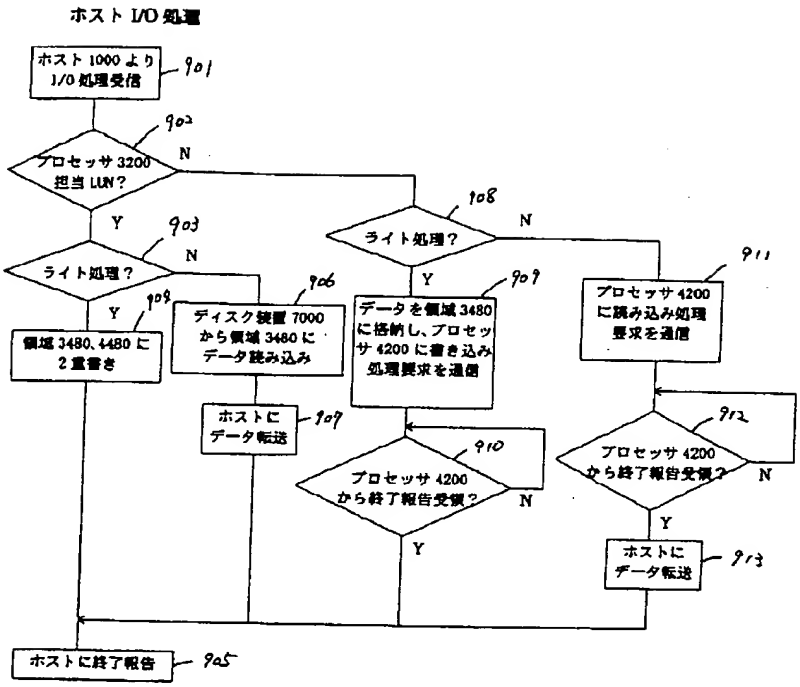
【図６】

図６



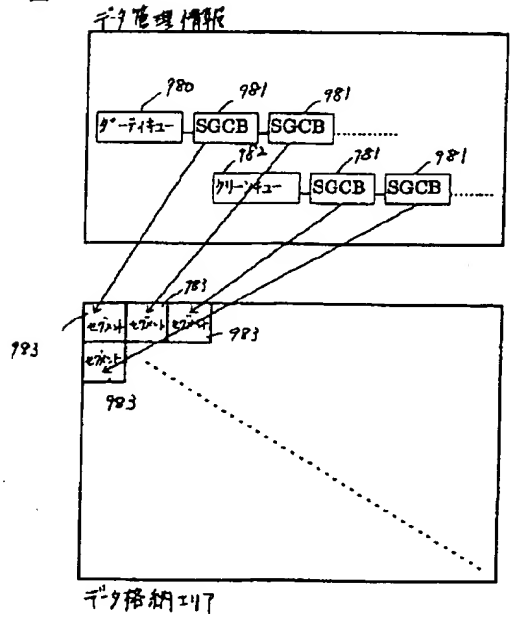
【図４】

図４


ホスト I/O 処理
ホスト 1000 より I/O 処理受信
901
プロセッサ 3200 担当 LUN?
902
N
Y
ライト処理?
903
N
Y
領域 3480、4480 に 2 重書き
904
ディスク装置 7000 から領域 3480 にデータ読み込み
906
ホストにデータ転送
907
ホストに終了報告
905
ライト処理?
908
N
Y
データを領域 3480 に格納し、プロセッサ 4200 に書き込み処理要求を通信
909
プロセッサ 4200 から終了報告受領?
910
N
Y
プロセッサ 4200 に読み込み処理要求を通信
911
プロセッサ 4200 から終了報告受領?
912
N
Y
ホストにデータ転送
913

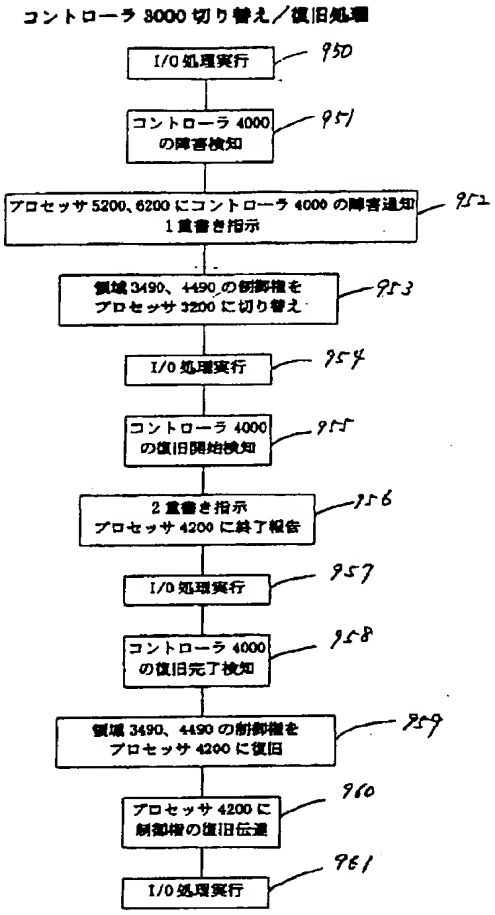
【図９】

図９


データ管理情報
980
981
981
SGCB
SGCB
982
981
981
SGCB
SGCB
983
983
993
993
データ格納エリア

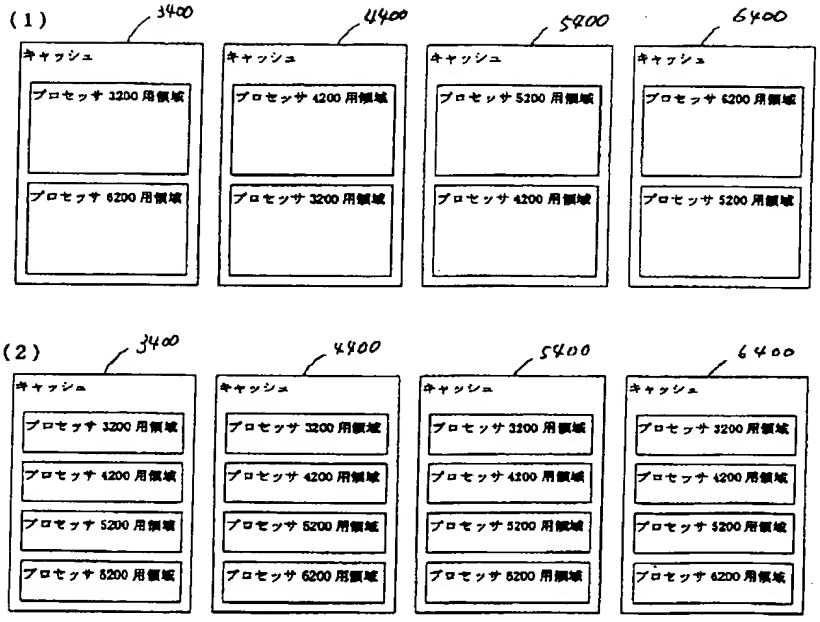
【図７】

図７


コントローラ 3000 切り替え／復旧処理
I/O処理実行
950
コントローラ 4000 の障害検知
951
プロセッサ 5200、6200 にコントローラ 4000 の障害通知 1 重書き指示
952
領域 3490、4490 の制御権を プロセッサ 3200 に切り替え
953
I/O処理実行
954
コントローラ 4000 の復旧開始検知
955
2 重書き指示 プロセッサ 4200 に終了報告
956
I/O処理実行
957
コントローラ 4000 の復旧完了検知
958
領域 3490、4490 の制御権を プロセッサ 4200 に復旧
959
プロセッサ 4200 に 制御権の復旧伝達
960
I/O処理実行
961

【図１０】

図１０


(1)
3400
キャッシュ
プロセッサ 3200 用領域
プロセッサ 6200 用領域
4400
キャッシュ
プロセッサ 4200 用領域
プロセッサ 3200 用領域
5400
キャッシュ
プロセッサ 5200 用領域
プロセッサ 4200 用領域
6400
キャッシュ
プロセッサ 6200 用領域
プロセッサ 5200 用領域
(2)
3400
キャッシュ
プロセッサ 3200 用領域
プロセッサ 4200 用領域
プロセッサ 5200 用領域
プロセッサ 6200 用領域
4400
キャッシュ
プロセッサ 3200 用領域
プロセッサ 4200 用領域
プロセッサ 5200 用領域
プロセッサ 6200 用領域
5400
キャッシュ
プロセッサ 3200 用領域
プロセッサ 4200 用領域
プロセッサ 5200 用領域
プロセッサ 6200 用領域
6400
キャッシュ
プロセッサ 3200 用領域
プロセッサ 4200 用領域
プロセッサ 5200 用領域
プロセッサ 6200 用領域